



NIT-318

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Yoshifumi TAKAMOTO et al.
Serial No.: 10/046,203
Filed: January 16, 2002
For: METHOD FOR CORRECTING A PROGRAM RUNNING ON A
COMPUTER SYSTEM
Group: 2113
Examiner: A. Riad

CLAIM FOR PRIORITY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

May 9, 2006

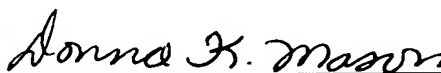
Sir:

Under the provisions of 35 USC §119 and 37 CFR §1.55, Applicants hereby claim the right of priority based on Japanese Patent Application No. 2001-130919, filed in Japan on April 27, 2001.

A certified copy of the Japanese Patent Application is attached hereto.

Respectfully submitted,

MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C.



Donna K. Mason
Registration No. 45,962

DKM/sdb
(703) 684-1120



PATENT OFFICE

JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following application as filed with this office.

Date of Application : April 27, 2001
Application Number : Patent Application No. 130919 of 2001
Applicant (s) : Hitachi, Ltd.

Dated this 11th day of January, 2002

Kouzou OIKAWA
Commissioner,
Patent Office
Certificate No. 2001-3113480

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日
Date of Application:

2001年 4月27日

出 願 番 号
Application Number:

特願2001-130919

ST.10/C]:

[JP2001-130919]

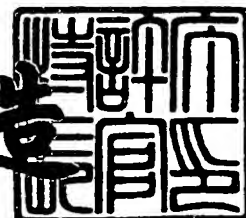
出 願 人
Applicant(s):

株式会社日立製作所

2002年 1月11日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



CERTIFIED COPY OF
PRIORITY DOCUMENT

出証番号 出証特2001-3113480

BEST AVAILABLE COPY

【書類名】 特許願

【整理番号】 NT01P0124

【提出日】 平成13年 4月27日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/46

【発明者】

 【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

 【氏名】 高本 良史

【発明者】

 【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

 【氏名】 佐川 暢俊

【発明者】

 【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

 【氏名】 鶴飼 敏之

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社日立製作所

【代理人】

 【識別番号】 100068504

 【弁理士】

 【氏名又は名称】 小川 勝男

 【電話番号】 03-3661-0071

【選任した代理人】

 【識別番号】 100086656

 【弁理士】

 【氏名又は名称】 田中 恭助

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100094352

【弁理士】

【氏名又は名称】 佐々木 孝

【電話番号】 03-3661-0071

【手数料の表示】

【予納台帳番号】 081423

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラム修正方法

【特許請求の範囲】

【請求項 1】

複数の論理計算機を有する計算機システムについて、稼動中の第 1 の論理計算機の制御下のプログラムとデータを記憶手段から記憶手段へコピーし、コピーされた前記プログラムを用いて待機中の第 2 の論理計算機を起動し、コピーされた前記プログラム中の修正プログラムを用いてコピーされた前記プログラムを修正し、前記第 1 の論理計算機の制御下のネットワークが前記第 2 の論理計算機の制御下になるように切り替えを行い、前記第 1 の論理計算機の稼動から修正された前記プログラムを用いる前記第 2 の論理計算機の稼動に切り替えることを特徴とするプログラム修正方法。

【請求項 2】

前記コピーを実行時に、コピー元の前記記憶手段に施される更新の履歴を採取し、コピー終了後に前記履歴を用いてコピー先の前記記憶手段を更新することを特徴とする請求項 1 記載のプログラム修正方法。

【請求項 3】

コピーされた前記プログラムの修正の結果を前記第 2 の論理計算機から前記第 1 の論理計算機に通知し、正常と判定される場合に前記第 1 の論理計算機の稼動からコピーされ修正された前記プログラムを用いる前記第 2 の論理計算機の稼動に切り替えることを特徴とする請求項 1 記載のプログラム修正方法。

【請求項 4】

前記ネットワークの切り替えの際に、前記ネットワークを介する外部要求の受付禁止とし、すでに受け付けられた外部要求の処理を完了し、前記第 1 の論理計算機の制御下の第 1 のネットワーク装置の第 1 のネットワークアドレスを第 2 のネットワークアドレスに変更し、前記第 2 の論理計算機の制御下の第 2 のネットワーク装置のネットワークアドレスとして前記第 1 のネットワークアドレスを設定することを特徴とする請求項 1 記載のプログラム修正方法。

【請求項 5】

稼働中の前記第 1 の論理計算機の制御下の第 1 のネットワーク装置のネットワークアドレスが第 1 のネットワークアドレスである場合に、前記ネットワークの切り替えに際して、前記第 2 の論理計算機の制御下の第 2 のネットワーク装置のネットワークアドレスとして前記第 1 のネットワークアドレスを設定し、前記第 1 のネットワーク装置のネットワークアドレスを第 2 のネットワークアドレスに変更し、前記ネットワークを介して外部に送信するパケットの送信元アドレスが前記第 2 のネットワークアドレスであるパケットについて前記送信元アドレスを前記第 1 のネットワークアドレスに変更することを特徴とする請求項 1 記載のプログラム修正方法。

【請求項 6】

外部に修正プログラムの新規登録があるか否かを監視し、前記修正プログラムが新規登録された場合に登録された前記修正プログラムをダウンロードして記憶手段に登録し、前記修正プログラムの登録を外部の管理者端末に通知し、前記管理者端末からの指示に従って前記修正プログラムを実行して修正対象のプログラムを修正することを特徴とするプログラム修正方法。

【請求項 7】

コンピュータに、複数の論理計算機を有する計算機システムの中で稼働中の第 1 の論理計算機の制御下のプログラムとデータを記憶手段から記憶手段へコピーする機能、コピーされた前記プログラムを用いて待機中の第 2 の論理計算機を起動する機能、コピーされた前記プログラム中の修正プログラムを用いてコピーされた前記プログラムを修正する機能、前記第 1 の論理計算機の制御下のネットワークが前記第 2 の論理計算機の制御下になるように切り替えを行う機能、および前記第 1 の論理計算機の稼働から修正された前記プログラムを用いる前記第 2 の論理計算機の稼働に切り替える機能を実現させるためのプログラム。

【請求項 8】

さらに前記コピーを実行時に、コピー元の前記記憶手段に施される更新の履歴を採取する機能、およびコピー終了後に前記履歴を用いてコピー先の前記記憶手段を更新する機能を実現させることを特徴とする請求項 7 記載のプログラム。

【請求項 9】

さらにコピーされた前記プログラムの修正の結果を前記第 2 の論理計算機から前記第 1 の論理計算機に通知する機能、および正常と判定される場合に前記第 1 の論理計算機の稼動からコピーされ修正された前記プログラムを用いる前記第 2 の論理計算機の稼動に切り替える機能を実現させることを特徴とする請求項 7 記載のプログラム。

【請求項 1 0】

さらに前記ネットワークの切り替えの際に、前記ネットワークを介する外部要求の受付禁止とする機能、すでに受け付けられた外部要求の処理を完了する機能、前記第 1 の論理計算機の制御下の第 1 のネットワーク装置の第 1 のネットワークアドレスを第 2 のネットワークアドレスに変更する機能、および前記第 2 の論理計算機の制御下の第 2 のネットワーク装置のネットワークアドレスとして前記第 1 のネットワークアドレスを設定する機能を実現させることを特徴とする請求項 7 記載のプログラム。

【請求項 1 1】

さらに稼動中の前記第 1 の論理計算機の制御下の第 1 のネットワーク装置のネットワークアドレスが第 1 のネットワークアドレスである場合に、前記ネットワークの切り替えに際して、前記第 2 の論理計算機の制御下の第 2 のネットワーク装置のネットワークアドレスとして前記第 1 のネットワークアドレスを設定する機能、前記第 1 のネットワーク装置のネットワークアドレスを第 2 のネットワークアドレスに変更する機能、および前記ネットワークを介して外部に送信するパケットの送信元アドレスが前記第 2 のネットワークアドレスであるパケットについて前記送信元アドレスを前記第 1 のネットワークアドレスに変更する機能を実現させることを特徴とする請求項 7 記載のプログラム。

【請求項 1 2】

コンピュータに、外部に修正プログラムの新規登録があるか否かを監視する機能、前記修正プログラムが新規登録された場合に登録された前記修正プログラムをダウンロードして記憶手段に登録する機能、前記修正プログラムの登録を外部の管理者端末に通知する機能、および前記管理者端末からの指示に従って前記修正プログラムを実行して修正対象のプログラムを修正する機能を実現させること

を特徴とするプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、計算機上で稼動しているプログラムの修正方法に係わり、特にサービスを停止させることなくプログラムの修正を行う方法に関する。

【0002】

【従来の技術】

インターネットの普及により、あらゆる計算機が実質的に世界中のどこからでもアクセスできるようになった。しかし同時に計算機が不正にアクセスされるケースも増大することが予想される。こういった事態に対し、計算機が不正にアクセスされないようにするパッチファイルが公開されている。パッチファイルとはプログラムを修正するためのプログラム変更情報が記録されたファイルであり、このパッチファイルを使用しプログラムの修正を行うことによって不正アクセスを防止することができる。パッチファイルの適用は、計算機上で稼動している全てのプログラムが対象となる。例えば、オペレーティングシステムやアプリケーションなどである。パッチファイルを用いたプログラム修正手順は次の通りである。(1) サービスの停止、(2) パッチファイルを適用しプログラムを修正し、(3) システムを再起動し、(4) サービスを再開する。このようにプログラムを修正するためには、一旦サービスを停止する必要がある。これは、通常オペレーティングシステムやアプリケーションを稼動したままプログラムの修正が行えないためである。そのため、プログラムの修正は一時的に使用中のプログラムとは別のプログラムに対して適用し、次回起動時には修正されたプログラムを使って起動する手段を用いている。

【0003】

【発明が解決しようとする課題】

従来のプログラム修正方法では、サービスを一旦停止し、プログラム修正を行う必要がある。プログラム修正のための新規のパッチファイルは頻繁に発生しており、そのたびにサービスを停止してはサービスの質を低下させることにな

る。一方、パッチファイルをある程度まとめてプログラム修正を行うことでサービスの停止を抑える運用方法もある。しかしいつ誰から計算機に対して不正アクセスをされるかわからないので、セキュリティに関するプログラム修正を引き伸ばすのは好ましくない。プログラムの修正に関する技術として、特開平 8 - 2 6 3 2 7 9 号公報に記載された技術がある。これは計算機を仮想化し、1 台の物理計算機から複数の論理的な計算機を構築する機構を持った計算機システムにおいて、論理計算機上で稼動しているプログラムを終了することなくプログラム修正を行う方法が述べられている。しかしそのプログラム修正は、論理計算機上で稼動しているプログラムに対してではなく、複数の論理的な計算機を集中的に制御する制御プログラムに対するプログラム修正である。またこの制御プログラムを修正している間は、ほとんどの論理計算機上のプログラムは停止している。つまり論理計算機上のプログラムは機能を停止させられているためサービスを継続することはできない。またサービス进行处理している論理計算機上のオペレーティングシステムやアプリケーションに対するプログラム修正ではない。

【 0 0 0 4 】

本発明の目的は、論理計算機上で稼動しているサービスを停止させることなくプログラム修正を行う方法を提供することにある。

【 0 0 0 5 】

本発明の他の目的は、外部で修正プログラムの新規登録が発生したことを迅速に検知し、その修正プログラムを用いて早急にプログラム修正を行う方法を提供することにある。

【 0 0 0 6 】

【課題を解決するための手段】

本発明は、複数の論理計算機を有する計算機システムについて、稼動中の第 1 の論理計算機の制御下のプログラムとデータを記憶手段から記憶手段へコピーし、コピーされたそのプログラムを用いて待機中の第 2 の論理計算機を起動し、コピーされたプログラム中の修正プログラムを用いてコピーされたそのプログラムを修正し、第 1 の論理計算機の制御下のネットワークが第 2 の論理計算機の制御下になるように切り替えを行い、第 1 の論理計算機の稼動から修正されたプログ

ラムを用いる第2の論理計算機の稼動に切り替えるプログラム修正方法の特徴とする。

【0007】

また本発明は、外部に修正プログラムの新規登録があるか否かを監視し、修正プログラムが新規登録された場合に登録された修正プログラムをダウンロードして記憶手段に登録し、その修正プログラムの登録を外部の管理者端末に通知し、管理者端末からの指示に従って修正プログラムを実行して修正対象のプログラムを修正するプログラム修正方法の特徴とする。

【0008】

【発明の実施の形態】

〔実施例1〕

以下、本発明に係る論理計算機環境におけるプログラム修正方法の実施例1について図面を用いて詳細に説明する。

【0009】

図1は、実施例1の論理計算機環境におけるプログラム修正方法の概略構成図を示している。101は物理計算機であり、102、103、104はそれぞれ物理計算機101内のプロセッサ、ディスク装置、ネットワーク装置である。110は計算機資源分割機構であり、物理計算機101から論理的な複数の計算機を構成するための機構である。105、106は計算機資源分割機構によって生成された論理計算機である。107、108、109は論理計算機105内で動作するプログラムであり、それぞれアプリケーションプログラム、オペレーティングシステム、パッチマネージャである。オペレーティングシステム108は、論理計算機105で稼動するゲストOSと呼ばれるオペレーティングシステムであり、論理計算機105のハードウェア制御やアプリケーション107の動作を支援する。パッチマネージャ109は、本発明の特徴となるプログラムであり、論理計算機105上で稼動しているアプリケーションが提供するサービスを停止させることなくプログラム修正を行うプログラムである。論理計算機106は、パッチ用の予備論理計算機（待機中の論理計算機）であり、待機中のためプログラムが動作していない状態を示している。論理計算機105は、複数存在可能で

ある。

【 0 0 1 0 】

図 2 は、パッチマネージャ 1 0 9 によるプログラム修正の流れを示している。1 0 5, 1 0 6 はそれぞれ論理計算機である。論理計算機 1 (1 0 5) 上で既にサービスが稼動しており (ステップ 2 0 3) 、論理計算機 2 (1 0 6) を使用し、サービスを停止させることなくプログラム修正を行う処理の流れを示している。ステップ 2 0 4 ではパッチマネージャ 1 0 9 がパッチ処理の起動をする。この起動処理中に論理計算機 1 (1 0 5) 内のプログラム及びデータが論理計算機 2 (1 0 6) 内にコピーされる。次に論理計算機 2 (1 0 6) 上でパッチ処理が実行され、論理計算機 2 (1 0 6) 上のソフトウェアのプログラム修正が行われる (ステップ 2 0 5) 。論理計算機 2 (1 0 6) 上のソフトウェアのプログラム修正が行われた後、論理計算機 1 (1 0 5) に制御が移行し、論理計算機 1 (1 0 5) はシステム切り替え処理 (ステップ 2 0 6) を実行する。システム切り替え処理 (ステップ 2 0 6) により、サービスは論理計算機 1 (1 0 5) の制御を離れ、論理計算機 2 (1 0 6) で稼動することになる (ステップ 2 0 7) 。これらの処理により、サービスはプログラムが修正された論理計算機上で稼動できるため、高信頼なサービスを提供できるようになるとともに、一時的に論理計算機を 2 つ使用することによってサービスを停止する必要がなくなる。

【 0 0 1 1 】

図 3 は、パッチマネージャ 1 0 9 の構成を示している。パッチマネージャ 1 0 9 は、パッチ実行プログラム 3 0 2、パッチ監視プログラム 3 0 3、論理計算機資源管理テーブル 3 0 6、パッチテーブル 3 0 7、論理計算機制御プログラム 3 0 5 から構成される。パッチ実行プログラム 3 0 2 は、予備論理計算機上でパッチ処理を起動するプログラムである。パッチ監視プログラム 3 0 3 は、インターネット上等のパッチファイルが格納されたサーバ (図示せず) を定期的に監視し、新しいパッチファイルが登録されたかどうかを監視し、パッチ処理を管理者に促すプログラムである。論理計算機資源管理テーブル 3 0 6 は、各論理計算機が持つプロセッサ、ディスク装置、ネットワーク装置等の資源が登録されている。パッチテーブル 3 0 7 はパッチファイルを管理するテーブルで、既に処理したパ

ッチファイルと未だ処理していないパッチファイルの区分等の情報が登録されている。論理計算機資源管理テーブル306とパッチテーブル307は、パッチ実行プログラム302やパッチ監視プログラム303の実行時に参照／更新される。論理計算機制御プログラム305は、パッチ実行プログラム302やパッチ監視プログラム303が他の論理計算機を制御する場合に使用される。論理計算機制御プログラム305は、論理計算機間の通信や論理計算機の起動などのインターフェースを提供する。

【0012】

図4は、稼動中の論理計算機の一般的な構成を示している。論理計算機105内には、サービスを提供するアプリケーションプログラム107、オペレーティングシステム108、パッチマネージャ109がある。またアプリケーションプログラム107が使用するデータを格納するアプリケーションディスク405やオペレーティングシステム108が使用するシステムディスク406、外部とネットワーク通信するためのネットワーク装置104が接続されている。本実施例では、図4に示された論理計算機の構成を例にサービスを停止させないプログラム修正方法について述べる。

【0013】

図5は、論理計算機資源管理テーブル306の構造を示している。論理計算機資源管理テーブル306は、各論理計算機が所有するプロセッサ、ディスク装置、ネットワーク装置、メモリ容量等の資源割当情報を格納する。カラム501は論理計算機の識別子、カラム502はプロセッサ割当率、カラム503は割り当てられたメモリ容量、カラム504は割り当てられたディスク装置、カラム505は割り当てられたネットワーク装置を示している。例えば、論理計算機1は、物理プロセッサ1を100%割り当てられており、周辺装置としてDISK1およびDISK2、またネットワーク装置NET1が割り当てられていることを示している。この例では、論理計算機1又は論理計算機2が他の論理計算機に割り当てられた資源をカバーする資源を割り当てられており、他の論理計算機について予備論理計算機となり得る。

【0014】

図 6 は、パッチテーブル 3 0 7 の構造を示している。パッチテーブル 3 0 7 に
は、これまでに処理したパッチファイル情報や未だ処理していないパッチファイ
ルの情報などが格納されている。カラム 6 0 1 はパッチファイル名、カラム 6 0
2 からカラム 6 0 4 はパッチファイルの種類を示している。カラム 6 0 2 はセキ
ュリティ関連、カラム 6 0 3 は機能向上関連、カラム 6 0 4 は機能修正関連など
である。カラム 6 0 5 は修正された日時を示している。カラム 6 0 6 は、次の
パッチ処理時にパッチを適用するかどうか記述されている。カラム 6 0 7 は、
最後に処理されたパッチ処理実行結果が格納されている。

【 0 0 1 5 】

図 7 は、パッチ実行プログラム 3 0 2 のフローチャートを示している。ステッ
プ 7 0 1 は、予備論理計算機のサーチを行う。このステップは、パッチ処理をし
たい論理計算機と同等な資源を持つ未使用論理計算機をサーチする。このために
、図 5 に示す論理計算機資源管理テーブル 3 0 6 を参照し、例えば論理計算機 1
のプログラムにパッチ処理を行う場合、割り当てられた資源をカバーする論理計
算機 2 を割り当てる。ステップ 7 0 2 はシステムディスクをチェックする。ここ
でシステムディスクとは現在サービスを実行中の論理計算機が使用しているオペ
レーティングシステムを格納する記憶装置である。主なチェック項目は、システ
ムがウイルス等の不正なプログラムを所有していないかどうかのチェックである
。これは、ステップ 7 0 3 でシステムディスクのコピーを行うに当たっての前処
理であり、不正なプログラムが格納された状態のまま、新しいシステムに移行す
るのを防止するのが目的である。ステップ 7 0 3 ではシステムディスクのコピー
を行う。例えば論理計算機資源管理テーブル 3 0 6 の論理計算機 1 の D I S K 1
がシステムディスクの場合、論理計算機 2 の D I S K 3 へディスクの内容をコピー
する。ステップ 7 0 4 ではアプリケーションディスクのコピーを行う。アプリ
ケーションディスクとは、アプリケーションプログラムやアプリケーションデー
タを格納する記憶装置である。このディスクには、現在サービスを行っている論
理計算機上で稼動しているアプリケーションが使用するデータが格納されている
。例えば論理計算機資源管理テーブル 3 0 6 の論理計算機 1 の D I S K 2 を論理
計算機 2 の D I S K 4 へコピーする。ステップ 7 0 5 ではパッチファイルのコピ

ーを予備論理計算機のシステムディスクに対して行う。この時、パッチテーブル 3 0 7 も同じようにコピーする。図 6 のパッチテーブル 3 0 7 に記載されているパッチファイル 6 0 1 がコピーの対象となる。このパッチファイルを利用して、予備論理計算機が起動された時にパッチ処理が行われる。ステップ 7 0 6 ではパッチ実行スクリプトのコピーを予備論理計算機のシステムディスクに対して行う。パッチ実行スクリプトは、後で詳細に説明するが予備論理計算機が起動された時に自動的にパッチ処理が行われるようにするための自動化スクリプトである。

【 0 0 1 6 】

ステップ 7 0 7 は、予備論理計算機の起動を要求する。要求は、論理計算機制御プログラム 3 0 5 に対して発行され、論理計算機制御プログラム 3 0 5 は、計算機資源分割機構 1 1 0 に対して指定された論理計算機の起動要求を行う。これにより、予備論理計算機はステップ 7 0 2 でコピーされたシステムディスクを用いてオペレーティングシステム 1 0 8 が起動する。予備論理計算機が起動されると、ステップ 7 0 6 でコピーされたパッチ実行スクリプトが自動的に起動され所定のパッチ処理を実行する。ステップ 7 0 8 では、予備論理計算機の再起動要求を発行する。要求は、論理計算機制御プログラム 3 0 5 に対して発行され、論理計算機制御プログラム 3 0 5 は、計算機資源分割機構 1 1 0 に対して指定された論理計算機の再起動を行う。これは、パッチ処理を有効にするためである。パッチ処理を終了した時点では、またパッチ処理の内容は有効になっておらず、システム再起動すると有効になる。そのため、ステップ 7 0 8 で再起動要求を発行し予備論理計算機のシステムを再起動する。ステップ 7 0 9 では、予備論理計算機上でのパッチ処理が正常に終了したかどうかをチェックする。その結果、予備論理計算機上でのパッチ処理が正常に終了していれば、ステップ 7 0 9 でネットワーク切り替え処理を行いサービスを処理する論理計算機を交替する。予備論理計算機上でのパッチ処理が正常に終了していなければ、現状の論理計算機上でサービスを継続して実行する。パッチ処理が正常に終了したかどうか、すぐには判断できないケースもある。例えば、パッチ処理後に数時間稼動した後、不具合が発生する場合である。この場合、本発明の実施例を適用し、元の論理計算機にサービスを再交代させることでサービスを継続させつつ、その間にパッチ処理による

不具合の修正等が可能である。

【 0 0 1 7 】

図 8 は、図 7 のステップ 7 0 3 におけるシステムディスクのコピーの詳細フローを示している。ステップ 8 0 1 では書き込み履歴の採取開始を行う。これはコピー中に更新された情報を記録する目的である。ステップ 8 0 2 では、システムディスクのコピーを実施する。ステップ 8 0 3 ではコピーが終了したかどうかをチェックする。ステップ 8 0 4 では、書き込み履歴の採取を終了する。ステップ 8 0 5 では、コピー中に更新された情報を、コピー先のシステムディスクに反映する。こうすることで、コピーされたシステムディスクを最新の状態に保つことができる。

【 0 0 1 8 】

図 9 は、図 7 のステップ 7 0 4 におけるアプリケーションディスクのコピーの詳細フローを示している。ステップ 9 0 1 では書き込み履歴の採取開始を行う。これは、コピー中に更新された情報を記録する目的である。ステップ 9 0 2 では、アプリケーションディスクのコピーを実施する。ステップ 9 0 3 ではコピーが終了したかどうかをチェックする。ステップ 9 0 4 では、書き込み履歴の採取を終了する。ステップ 9 0 5 では、コピー中に更新された情報を、コピー先のアプリケーションディスクに反映する。こうすることで、コピーされたアプリケーションディスクを最新の状態に保つことができる。

【 0 0 1 9 】

図 1 0 は、図 8 および図 9 における書き込み履歴の採取を説明する図である。書き込み操作が行われる場合（1 0 0 1）、書き込み操作を履歴取得プログラム 1 0 0 2 がトラップし、書き込みデータは例えばシステムディスクの場合はシステムディスク 1 0 0 3 へ書き込まれると共に、書き込み履歴ディスク 1 0 0 8 に、書き込み時間 1 0 0 4、書き込み位置 1 0 0 5、データ長 1 0 0 6、データ 1 0 0 7 を追記書きする。このデータをもとに、システムディスクあるいはアプリケーションディスクのコピー後に最新の情報に更新することができる。これにより、ディスク容量が大きくコピーに時間がかかる場合にも、コピー中およびコピー後に発生した変更をコピー先のディスクに反映することができるようになる。

従って予備論理計算機が本番稼動前にステップ805及びステップ905を実行し、未更新データをディスクに反映することができる。また他の方法として、ディスク装置が持つ機能を利用することもできる。ディスク装置がディスクのミラーリングを有し、一時的にミラーリングを解除し2つのディスクに分割する機能を持っている場合、この機能を使用することができる。ステップ703およびステップ704の要求でミラーリングを解除し、一つのディスクを現状のサービスを提供している論理計算機に割当て、もう一つを予備論理計算機に割り当てることによって、コピーと同等の機能を提供することができる。この機能は、一方のディスクに更新が発生した場合、再度ミラーリングの形態に戻す時に分割された両方のディスクのデータを一致させる機能があり、これにより最新の情報を確保したままコピーが完了したのと同じ動作をさせることができる。

【0020】

図11は、図7のステップ706でコピーされるパッチ実行スクリプト1107と関連するプログラムを示している。オペレーティングシステムは、通常起動時に自動的に実行される起動スクリプト1102～1106を格納しておくことができる。この機能を使用し、オペレーティングシステムが起動したときにパッチ処理を行うように設定することが、パッチ実行スクリプト1107の目的である。オペレーティングシステムが起動した時に自動的にスクリプトを実行するスクリプト実行プログラム1101は、所定の場所に格納された各起動スクリプト1102～1106を順番に実行する。このような起動スクリプトの1つとして、パッチ実行スクリプト1107を格納しておく。このパッチ実行スクリプト1107は、パッチファイル1109とパッチテーブル307を参照しながらパッチ処理を実行する。このような起動スクリプトの代わりにマシン語で表現された起動プログラムでもよいが、スクリプト言語で記載されたスクリプトであればシステム管理者による変更が容易となる。スクリプトの例としてシステムコマンド列から成るスクリプトなどがある。

【0021】

図12は、パッチ実行スクリプト1107の詳細フローを示している。ステップ1201では、パッチテーブル307を読み込む。ステップ1202では、パ

ッチテーブル 3 0 7 をサーチし、パッチ処理を行うパッチファイル 6 0 1 を決定する。これは、図 6 のカラム 6 0 6 の次回実行カラムを参照することで実現できる。ステップ 1 2 0 3 では、所定のパッチファイル 1 1 0 9 を読み込み、ステップ 1 2 0 4 でパッチプログラムを実行する。パッチプログラムは、通常オペレーティングシステムが標準で有しており、このプログラムと指定されたパッチファイル 1 1 0 9 とで、オペレーティングシステム、アプリケーションなどのプログラムの修正を行うことができる。ステップ 1 2 0 5 では、パッチプログラムの実行結果をパッチテーブル 3 0 7 のカラム 6 0 7 に格納する。ステップ 1 2 0 6 では、指定された全てのパッチファイルが実行されたかどうかをチェックし、全て処理されていないければ、ステップ 1 2 0 3 を実行し、全て実行されていればステップ 1 2 0 7 へ移行する。ステップ 1 2 0 7 では、論理計算機制御プログラム 3 0 5 を介して論理計算機 1 のパッチマネージャ 1 0 9 へパッチ処理が終了したことを通知する。この通知では、予備論理計算機のパッチテーブル 3 0 7 と共に、パッチ処理が終了したことを論理計算機 1 のパッチマネージャ 1 0 9 に伝える。これにより、パッチマネージャ 1 0 9 は、パッチ処理の終了と、パッチ処理が正常に行われたかどうかの情報を有するパッチテーブル 3 0 7 からパッチ処理の結果を知ることができる。

【 0 0 2 2 】

図 1 3 は、ネットワークの切り替えフローを示している。ステップ 1 3 0 1 では、ネットワークの切り替えが完了するまでネットワークの外部要求の受付を禁止する。既に受け付けた要求は処理するが、新たな要求は一時的に受け付けなくする。ステップ 1 3 0 2 では、既に受け付けた要求を全て処理し終えるまで待機する。ステップ 1 3 0 3 では、ネットワークの切り替え要求を発行する。このネットワーク切り替え要求は、論理計算機制御プログラム 3 0 5 を介して計算機資源分割機構 1 1 0 に対して発行され、これを受けた計算機資源分割機構 1 1 0 は、ネットワークのハードウェア経路を現論理計算機から予備論理計算機に変更する。ステップ 1 3 0 4 では、ネットワークの切り替えが正常に行われ、論理計算機制御プログラム 3 0 5 を介して予備論理計算機のアプリケーションプログラム 1 0 7 を起動した後、サービスが正常に開始されたかどうかをチェックする。も

しネットワークやサービスの切り替えが正常に行われていない場合は、逆の手順によって元の論理計算機でサービスを再切替することで、サービスの停止を無くすることができる。ステップ 1 3 0 1 からしばらくの間、ネットワークの受付が一時的にできなくなる期間ができるが、ネットワークは再要求を自動的に発行するため、使用者にはこの影響はない。このネットワークの切り替えを図示したのが、図 1 4 である。

【 0 0 2 3 】

図 1 4 は、ネットワークの切り替えを説明する図である。この例では、論理計算機 1 (1 0 5) から論理計算機 2 (1 0 6) へネットワーク装置 1 0 4 を切り替える様子を示している。この機能は計算機資源分割機構 1 1 0 が有しており、ネットワーク切り替え要求 1 4 0 2 によってネットワーク装置 1 0 4 の通信経路が元の論理計算機の経路 1 4 0 3 から新しい論理計算機の経路 1 4 0 4 に切り替えられ、これによって、ネットワーク装置 1 0 4 のデータは論理計算機 2 (1 0 6) で受け付けることができるようになる。これにより、物理計算機 1 0 1 内の一つのネットワーク装置 1 0 4 の割当を変更することができるようになる。

【 0 0 2 4 】

図 1 5 は、物理計算機が 2 つ以上のネットワーク装置 1 0 4 を有していた場合の、ネットワーク切り替えフローを示している。図 1 3 で示したネットワーク切り替えは、一つのネットワーク装置 1 0 4 を切り替えるフローであるが、図 1 5 では、2 つ以上のネットワーク装置 1 0 4 を使用したネットワーク切り替えフローを示している。ステップ 1 5 0 1 では、ネットワークの切り替えが完了するまでネットワークの外部要求の受付を禁止する。既に受け付けた要求は処理するが、新たな要求は一時的に受け付けなくする。ステップ 1 5 0 2 では、既に受け付けた要求を全て処理し終えるまで待機する。ステップ 1 5 0 3 では、自ネットワークアドレスを変更する。この変更はオペレーティングシステム 1 0 8 が標準で有しているネットワーク情報変更プログラムによって行う。ステップ 1 5 0 4 では論理計算機制御プログラム 3 0 5 を介して予備論理計算機にアドレスの変更要求を発行する。ここでは、これまでサービスを処理していた論理計算機が使用していたネットワークアドレスを予備論理計算機に割り当てる。図 1 6 にこの処理

を示している。

【 0 0 2 5 】

図 1 6 では、ステップ 1 6 0 1 で予備論理計算機がネットワークアドレス変更要求を受け付ける。ステップ 1 6 0 2 ではネットワークアドレスを変更する。この変更はオペレーティングシステムが標準で有しているネットワーク情報変更プログラムによって行う。ステップ 1 6 0 3 では、ネットワークアドレスが変更されたことを、ネットワーク上の装置に通知する。ステップ 1 6 0 4 では、論理計算機制御プログラム 3 0 5 を介して切り替えが正常に行われたかどうかを要求元に通知する。これにより、予備論理計算機のネットワークアドレスの変更を行う。一方、図 1 5 のステップ 1 5 0 5 では、ネットワークアドレスが変更されたことをネットワーク上の装置に通知する。ステップ 1 5 0 6 では、ネットワークの切り替えが正常に行われ、サービスが正常に開始されたかどうかをチェックする。もしネットワークやサービスの切り替えが正常に行われていない場合は、逆の手順によって元の論理計算機でサービスを再切替することで、サービスの停止を無くすことができる。これらの処理により、これまでサービスを処理していた論理計算機は新しいネットワークアドレスを有し、予備論理計算機はこれまでサービスを提供していた論理計算機のネットワークアドレスを引き継ぐことができる。

【 0 0 2 6 】

図 1 7 は、物理計算機 1 0 1 が 2 つのネットワーク装置（1 0 4 - 1， 1 0 4 - 2）を有している場合のネットワークの切り替えを示している。この例では、論理計算機 1（1 0 5）から論理計算機 2（1 0 6）のネットワークアドレスの切り替えを示している。論理計算機 1（1 0 5）にはネットワーク装置 1 0 4 - 1 が割り当てられており、論理計算機 2 にはネットワーク装置 1 0 4 - 2 が割り当てられている。図 1 5 の処理により、例えば論理計算機 1 の旧ネットワークアドレス 1 1 1 1 1 1 1 1 を新しいネットワークアドレス（2 2 2 2 2 2 2 2）に変更し、論理計算機 2 のネットワークアドレスを 1 1 1 1 1 1 1 1 に変更する。論理計算機 1（1 0 5）が計算機資源分割機構 1 1 0 を介して論理計算機 2（1 0 6）へネットワークアドレス変更要求を発行することによって（1 7 0 3）、

ネットワーク装置が無くても論理計算機間で通信することができる。これらの処理により、2つの論理計算機のネットワーク接続を維持したまま、新しい論理計算機上にサービスを移行することができる。

【0027】

図18は、物理計算機101上の論理計算機105で稼動しているパッチマネージャ109内のパッチ監視プログラム303の概要を示している。パッチ監視プログラム303の機能は、インターネット1905上のパッチファイルが格納されたパッチサーバ1907の更新状況をチェックし、パッチファイルの更新が行われるとメールサーバ1911を介して管理者端末1906に通知する機能と、管理者端末1906が通知されたパッチの処理を行うかどうかをパッチ監視プログラム303に指示する機能を有する。パッチサーバ1907は、通常WWWサーバ1908またはファイル転送サーバで構成され、ディスク内にはパッチファイルや更新日付およびパッチの内容が記述されたファイル1910が格納されている。以下、パッチ監視プログラム303の詳細説明を行う。

【0028】

図19は、パッチ監視プログラム303のフローを示している。ステップ1801では、パッチ監視プログラム303がインターネット1905上等のパッチサーバ1907に接続する。ステップ1802ではそのパッチサーバを検索し、以前に監視した時間より後に更新されたパッチファイルがあればそれをダウンロードしてパッチテーブル307に登録する（ステップ1803）。ステップ1804では、新しく登録されたパッチファイルのリストを作成し、それをステップ1805で管理者端末1906に電子メールにて通知する。この時、パッチファイルの種別（図6のセキュリティ602から機能修正604）を参照し、特に重要なセキュリティに関する項目だけ管理者端末に送信するといったことも可能である。ステップ1806では、管理者端末1906からの電子メールによる指示を待つ。ステップ1807では、管理者端末1906からのメールの認証を行い、認証が正しければステップ1808に移行し、指示の内容に従いパッチテーブル307の次回実行606のカラムを更新する。その後、ステップ1809で、パッチ実行プログラム302を起動し、パッチ処理を行う。パッチ監視プログラ

ム 3 0 3 は定期的に起動し、新しくパッチファイルが登録されたかどうかを監視する。図 2 0 に管理者端末 1 9 0 6 とパッチ監視プログラム 3 0 3 の通信の概略を示す。

【 0 0 2 9 】

図 2 0 は、パッチ監視プログラム 3 0 3 と管理者端末 1 9 0 6 の電子メールの通信内容例を示している。新しくパッチファイルが登録されると、パッチ監視プログラム 3 0 3 は管理者端末 1 9 0 6 へ電子メールを送信する（ステップ 2 0 0 3）。この時、認証用のコードを送信する。管理者端末 1 9 0 6 は、電子メールを受けるとこれに対し、パッチ処理を行うかどうか記述された電子メールを、認証用コードから生成したコードと共にパッチ監視プログラム 3 0 3 に通知する（ステップ 2 0 0 4）。管理者かどうかの認証は、管理者端末 1 9 0 6 が生成したコードと同じ方法でパッチ監視プログラム 3 0 3 がコード生成し、管理者端末 1 9 0 6 が送信したコードと一致すれば管理者からの正しいメールと判断する。これらの電子メールは、それぞれメールサーバ 1 9 1 1 を介して行われる。上記により、頻繁に更新されるパッチ情報でも管理者は的確に判断でき、また電子メールで指示を出すことができるようになるため、遠隔地からでも即パッチを行うことができるようになる。

【 0 0 3 0 】

本実施例の効果として、物理的に一つのプロセッサしか持たない物理計算機 1 0 1 でも、計算機資源分割機構 1 1 0 を使用し、サービスを停止させないでパッチ処理を行うことができる。

【 0 0 3 1 】

また本実施例 1 ではハードウェアが有する計算機資源分割機構 1 1 0 を使用したが、ソフトウェアで計算機資源を分割し複数の仮想計算機を構築する場合でも適用可能である。しかし計算機資源の使用効率は実施例 1 で示したハードウェアによる計算機資源分割機構 1 1 0 の方が優れている。

〔実施例 2〕

本実施例 2 では、サービスの一時的な停止を無くするネットワーク切り替え方法を示している。実施例 1 では、図 1 3 のステップ 1 3 0 1 ～ 1 3 0 4 及び図 1

5のステップ1501～1506にて一時的にサービスを受け付けられない期間が存在した。これに対し、本実施例2では、全ての要求を受け付けられるようにすることが目的である。図21は、パッチマネージャ109'の構成を示している。パッチマネージャ109'は、パッチ実行プログラム302'、パッチ監視プログラム303、論理計算機資源管理テーブル306、パッチテーブル307、論理計算機制御プログラム305及びパケット変更プログラム2104から構成される。パッチ実行プログラム302'は、予備論理計算機を使用してプログラム修正処理を制御するプログラムである。パッチ監視プログラム303は、インターネット1905上等のパッチファイルが格納されたパッチサーバ1907を定期的に監視し、新しいパッチファイルが登録されたかどうかを監視し、パッチ処理を管理者に促すプログラムである。論理計算機資源管理テーブル306は、各論理計算機が持つプロセッサ、ディスク装置、ネットワーク装置等の資源が登録されている。パッチテーブル307はパッチファイルを管理するテーブルで、既に処理したパッチファイルと未だ処理していないパッチファイルの区分等が登録されている。論理計算機資源管理テーブル306とパッチテーブル307は、パッチ実行プログラム302'やパッチ監視プログラム303の実行時に参照／更新される。論理計算機制御プログラム305は、パッチ実行プログラム302'やパッチ監視プログラム303が論理計算機を制御する場合に使用される。論理計算機制御プログラム305は、論理計算機間の通信や論理計算機の起動などのインターフェースを提供する。パケット変更プログラム2104は、ネットワークに送信する情報を更新する機能を有し、これについては後で詳細に説明する。以下で説明する以外の各構成については、実施例1と同じである。

【0032】

図22は、パッチ実行プログラム302'内のネットワーク切り替えフローを示している。ステップ2201では、予備論理計算機にアドレスの変更要求を発行する。ステップ2202では、自ネットワークアドレスの変更を行う。ステップ2203では、ネットワークアドレスが変更されたことをネットワーク上の装置に通知する。この時点でこれまでサービスを処理していた論理計算機は新しいネットワークアドレスを有し、予備論理計算機はこれまでサービスを提供してい

た論理計算機のネットワークアドレスを引き継ぐことになる。新規に発行されたネットワークからの要求は、新しい予備論理計算機上で処理される。しかしこれまでサービスを処理していた論理計算機内には未処理の要求が残っており、これらを正常に処理する必要がある。そのためにステップ 2 2 0 4 でパッチ実行プログラム 3 0 2' は、パケット変更通知を発行する。これについては後で詳細に説明する。ステップ 2 2 0 5 では、ネットワークの切り替えが正常に行われ、サービスが正常に開始されたかどうかをチェックする。もしネットワークやサービスの切り替えが正常に行われていない場合は、逆の手順によって元の論理計算機でサービスを再切替することで、サービスの停止を無くすことができる。これらの処理により、これまでサービスを処理していた論理計算機は新しいネットワークアドレスを有し、予備論理計算機はこれまでサービスを提供していた論理計算機のネットワークアドレスを引き継ぐことができる。

【 0 0 3 3 】

図 2 3 は、パケット変更プログラム 2 1 0 4 の位置付けを示している。オペレーティングシステム 1 0 8 内のネットワーク制御プログラム 2 3 0 4 に、パケットを送信する処理を行うパケット送信プログラム 2 3 0 5 がある。パケット変更プログラム 2 1 0 4 は、このパケット送信プログラム 2 3 0 5 から呼び出されるプログラムである。

【 0 0 3 4 】

図 2 4 は、パケット送信プログラム 2 3 0 5 のフローを示している。ステップ 2 4 0 1 では、パケット送信プログラム 2 3 0 5 は、要求されたデータからパケットを生成する。ステップ 2 4 0 2 では、パケット変更通知が発行されたかどうかをチェックする。パケット変更通知は、図 2 2 のステップ 2 2 0 4 で発行される。もしパケット変更通知が発行されていれば、ステップ 2 4 0 3 に移行しパケット変更プログラム 2 1 0 4 を呼び出す。パケット変更プログラム 2 1 0 4 は、図 2 5 に示すように、パケット内の送信元アドレスをサービスを処理していた時に使用していた元のネットワークアドレスに変更する（ステップ 2 5 0 1）。ステップ 2 4 0 4 では、生成されたパケットを送信する。

【 0 0 3 5 】

図 2 6 は、パケットの構造を示している。パケットはネットワークを介して転送されるデータであり、送信元ネットワークアドレス 2 6 0 2 と送信先ネットワークアドレス 2 6 0 3 が格納されている。パケット変更プログラム 2 1 0 4 を実行する前のパケット構造は 2 6 0 4 で示されており、送信元ネットワークアドレス 2 6 0 2 には、図 2 2 のステップ 2 2 0 2 により新たに更新された新しいネットワークアドレスが格納される。しかしこのパケットを生成することになったネットワーク上の装置からの要求は、サービスを行っていたときに受け付けられており、このままその装置に返送すると、要求した装置はこのパケットを受け付けることができない。そのためパケット変更プログラム 2 1 0 4 は、パケット変更通知が発行されている間だけ、パケット 2 6 0 7 内の送信元アドレス 2 6 0 2 を、サービスを処理していた時に使用していた元のネットワークアドレスに変更する。こうすることで、要求の発行者は論理計算機が途中で変更されても問題が起これないようにすることができる。なお全ての要求が処理された時点でパケット変更通知は解除されることが望ましい。

【 0 0 3 6 】

実施例 2 の効果は、サービスをまったく停止させることなくパッチ処理を行うことができるようになることである。

【 0 0 3 7 】

また実施例 2 ではハードウェアが有する計算機資源分割機構 1 1 0 を使用したが、ソフトウェアで計算機資源を分割し複数の仮想計算機を構築する場合でも適用可能である。しかし計算機資源の使用効率は実施例 2 で示したハードウェアによる計算機資源分割機構 1 1 0 の方が優れている。

【 0 0 3 8 】

【発明の効果】

本発明により、サービスを停止させることなくパッチ処理を行えらるとともに、パッチファイルが頻繁に更新される場合でも管理者は容易にパッチファイルの更新を知ることができ、また遠隔地からでもパッチ更新の指示を出すことができる。

【図面の簡単な説明】

【図 1】

本発明の実施例 1 における全体構成図である。

【図 2】

実施形態のパッチ処理の流れを示す図である。

【図 3】

実施形態のパッチマネージャの構成を示す図である。

【図 4】

論理計算機の構成例を示す図である。

【図 5】

実施形態の論理計算機資源管理テーブルの構成を示す図である。

【図 6】

実施形態のパッチテーブルの構成を示す図である。

【図 7】

実施形態のパッチ実行プログラムのフロー図である。

【図 8】

実施形態のシステムディスクのコピーフロー図である。

【図 9】

実施形態のアプリケーションディスクのコピーフロー図である。

【図 10】

実施形態の書き込み履歴の構成を示す図である。

【図 11】

実施形態のスクリプトの構造を示す図である。

【図 12】

実施形態のパッチ実行スクリプトのフロー図である。

【図 13】

実施形態のネットワーク切り替えフロー図である。

【図 14】

実施形態のネットワーク切り替え処理を説明する図である。

【図 15】

実施形態のネットワーク切り替えフロー図である。

【図 1 6】

実施形態の予備論理計算機内のネットワーク切り替えフロー図である。

【図 1 7】

実施形態のネットワーク切り替え処理を説明する図である。

【図 1 8】

実施形態のパッチ監視プログラムの動作環境を示す図である。

【図 1 9】

実施形態のパッチ監視プログラムの処理フロー図である。

【図 2 0】

実施形態のパッチ監視プログラムの動作概要を示す図である。

【図 2 1】

実施例 2 におけるパッチマネージャの構成を示す図である。

【図 2 2】

実施例 2 におけるネットワーク切り替えフローを示す図である。

【図 2 3】

実施例 2 におけるパケット変更プログラムの位置付けを示す図である。

【図 2 4】

実施例 2 におけるパケット送信プログラムのフローを示す図である。

【図 2 5】

実施例 2 におけるパケット変更プログラムのフローを示す図である。

【図 2 6】

実施例 2 におけるパケットの構成を示す図である。

【符号の説明】

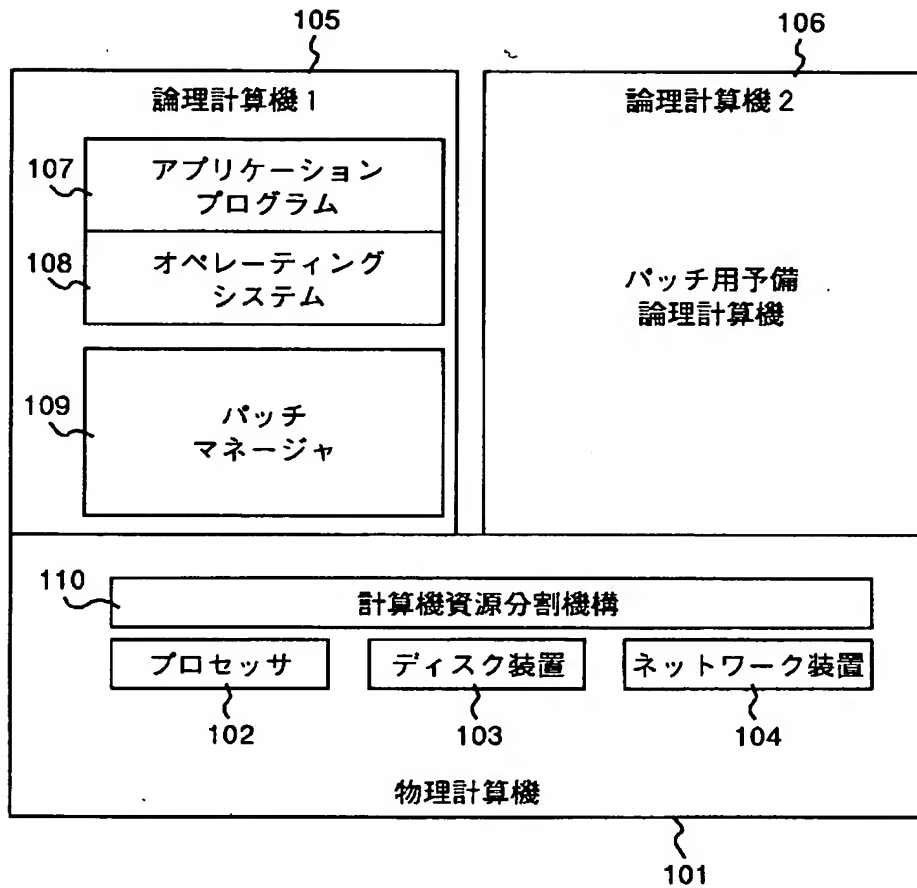
1 0 1 : 物理計算機、1 0 5, 1 0 6 : 論理計算機、1 0 7 : アプリケーションプログラム、1 0 8 : オペレーティングシステム、1 0 9 : パッチマネージャ、1 1 0 : 計算機資源分割機構、3 0 2 : パッチ実行プログラム、3 0 3 : パッチ監視プログラム、3 0 7 : パッチテーブル、1 1 0 7 : パッチ実行スクリプト、1 1 0 9 : パッチファイル、1 9 0 6 : 管理者端末、1 9 0 7 : パッチサーバ

、 1 9 1 1 : メールサーバ

【書類名】 図面

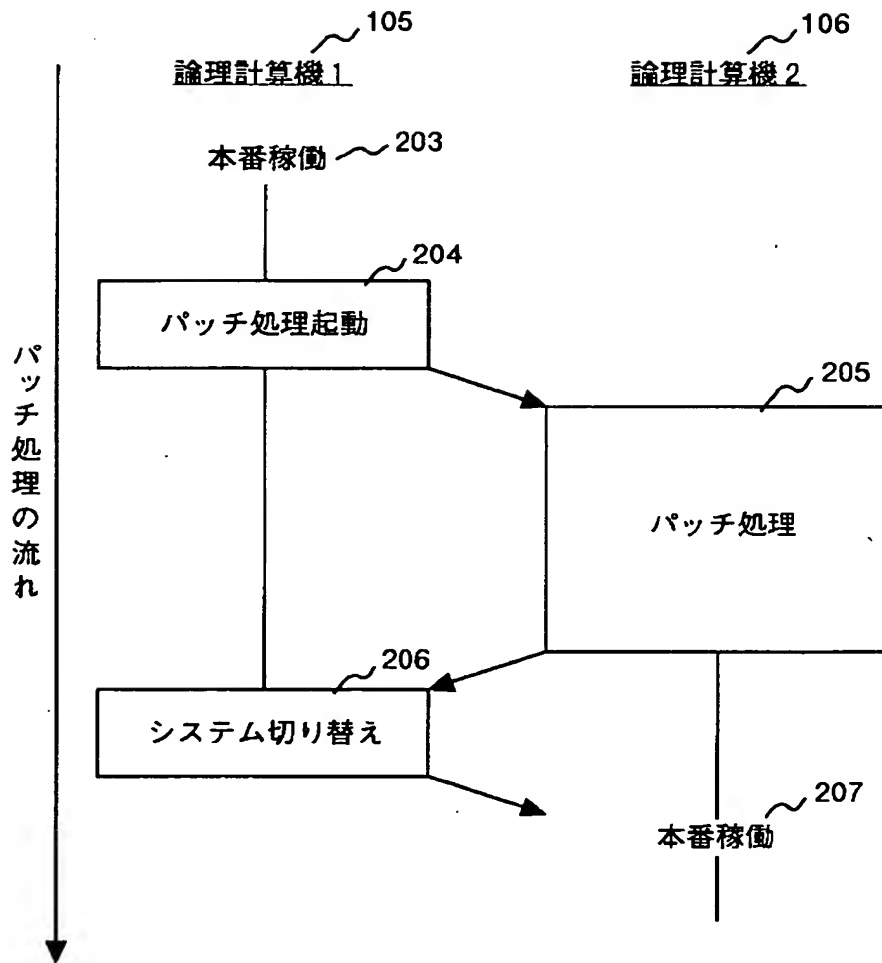
【図 1】

図 1



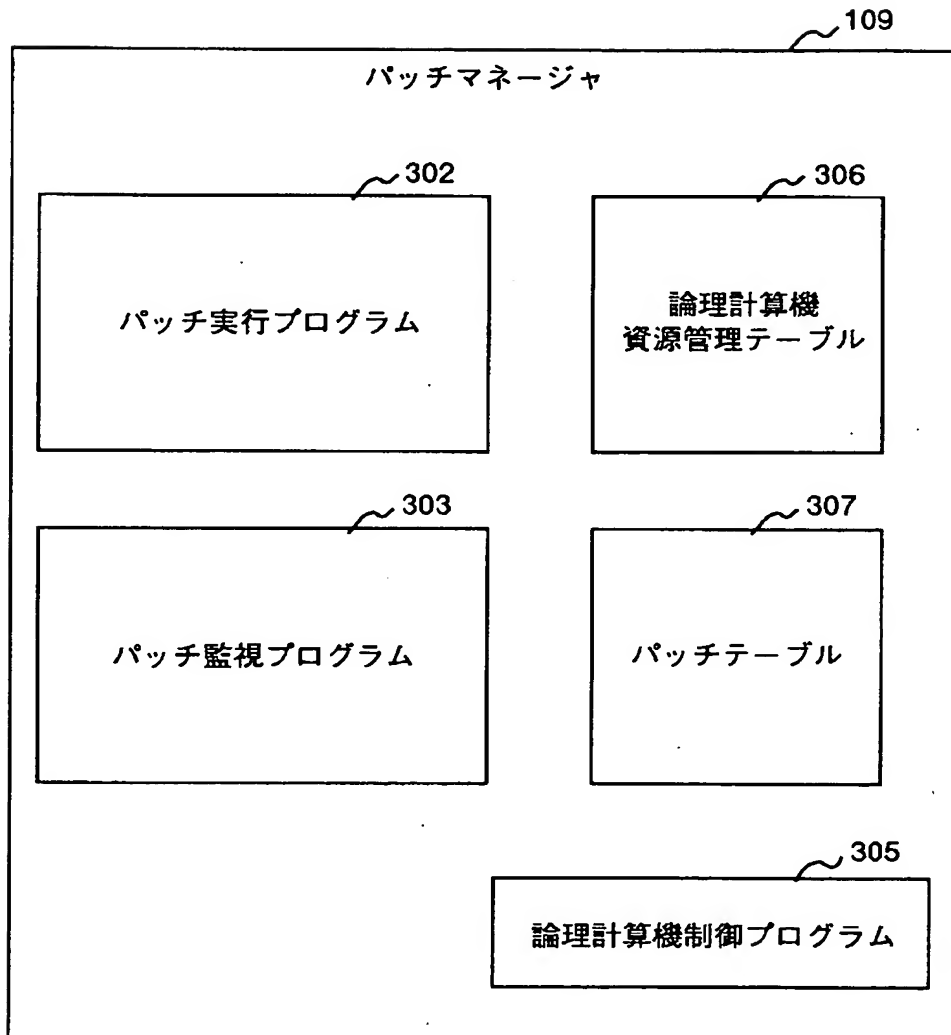
【図 2】

図 2



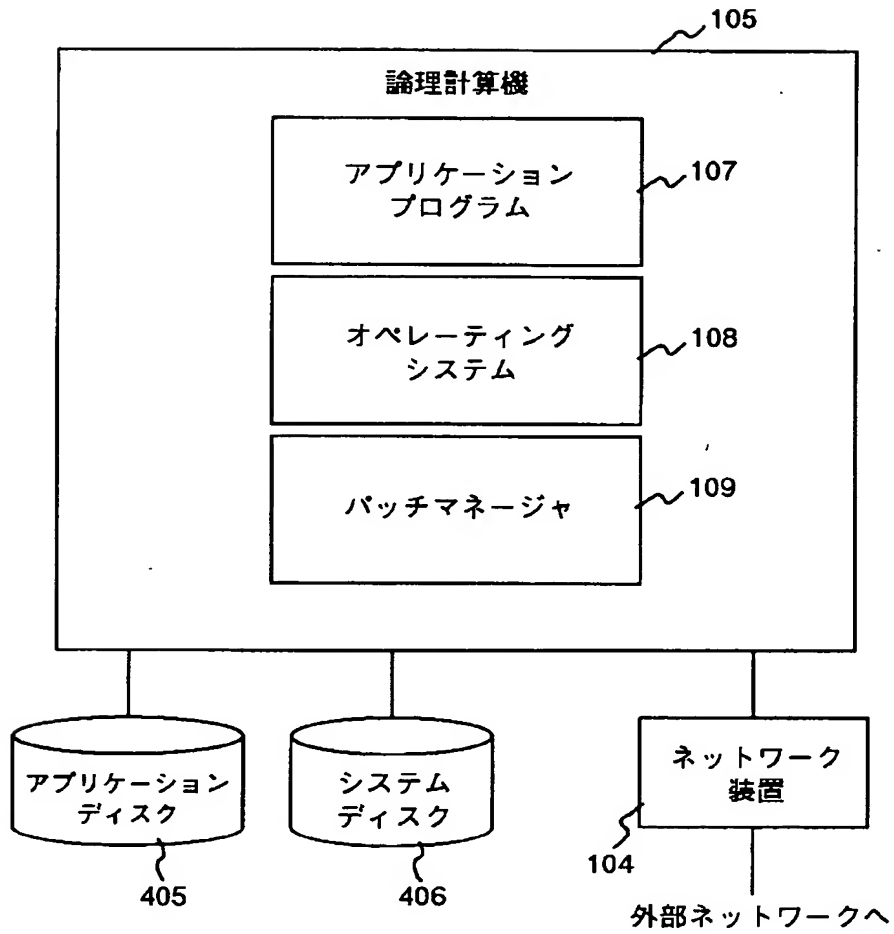
【図 3】

図 3



【図 4】

図 4



【図 5】

図 5

306：論理計算機資源管理テーブル

501 502 503 504 505				
論理計算機 ID	プロセッサ/ 割当率	メモリ	ディスク 装置	ネットワーク 装置
論理計算機 1	CPU1/100%	128MB	DISK1 DISK2	NET1
論理計算機 2	CPU2/100%	128MB	DISK3 DISK4	NET2
論理計算機 3	CPU3/50%	64MB	DISK5	
論理計算機 4	CPU3/50%	64MB	DISK6	NET3

【図 6】

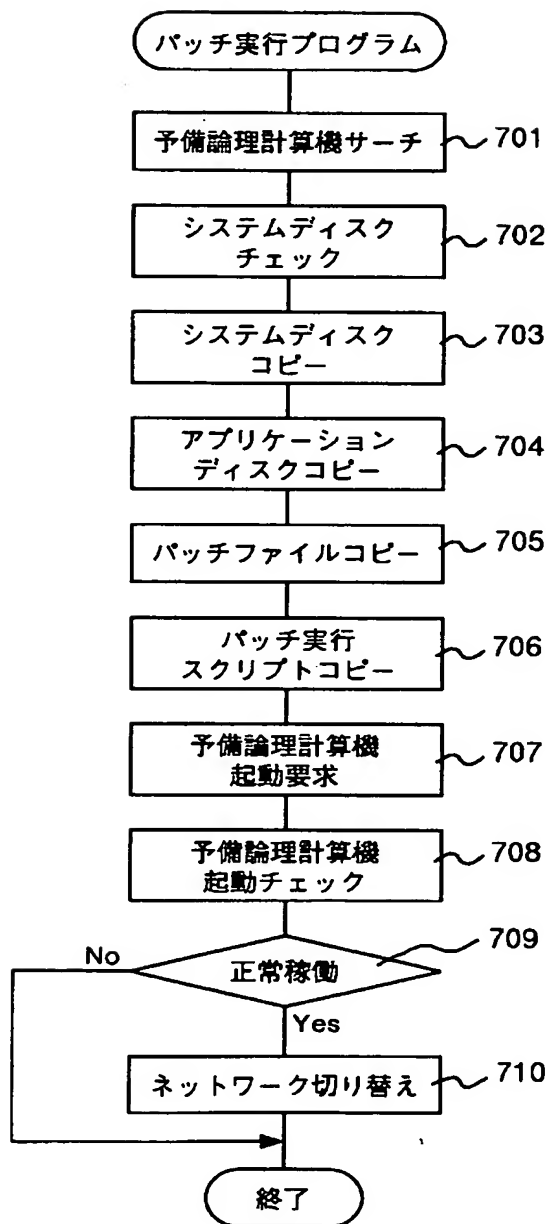
図 6

307：パッチテーブル

	601	602	603	604	605	606	607
パッチファイル	セキュリティ	機能向上	機能修正	修正	次回実行	実行結果	
パッチ 1	○			1998/12/25			
パッチ 2		○		未	×		
パッチ 3			○	未	×		
パッチ 4	○			1999/3/26			
パッチ 5	○			1999/8/15			
パッチ 6			○	1999/9/15			
パッチ 7	○			未	○		

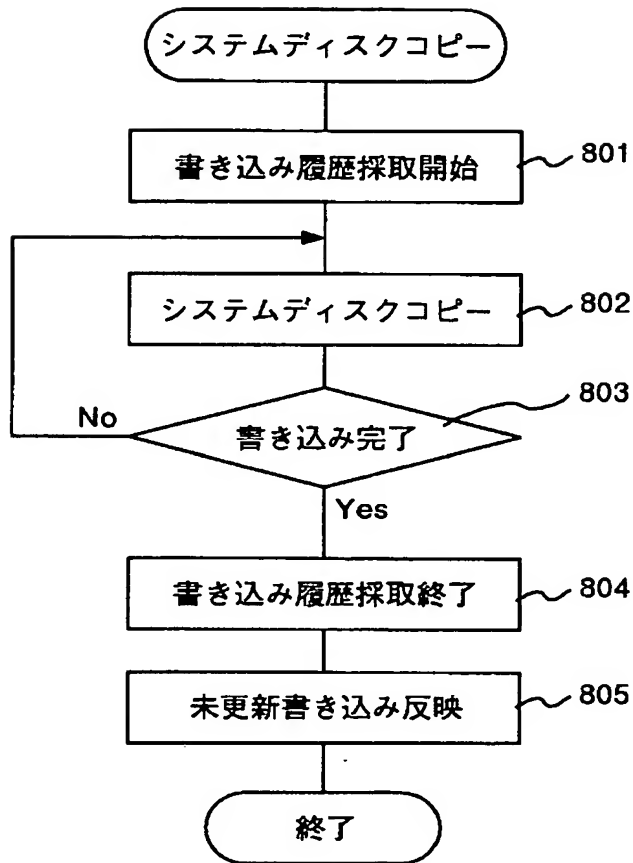
【図 7】

図 7



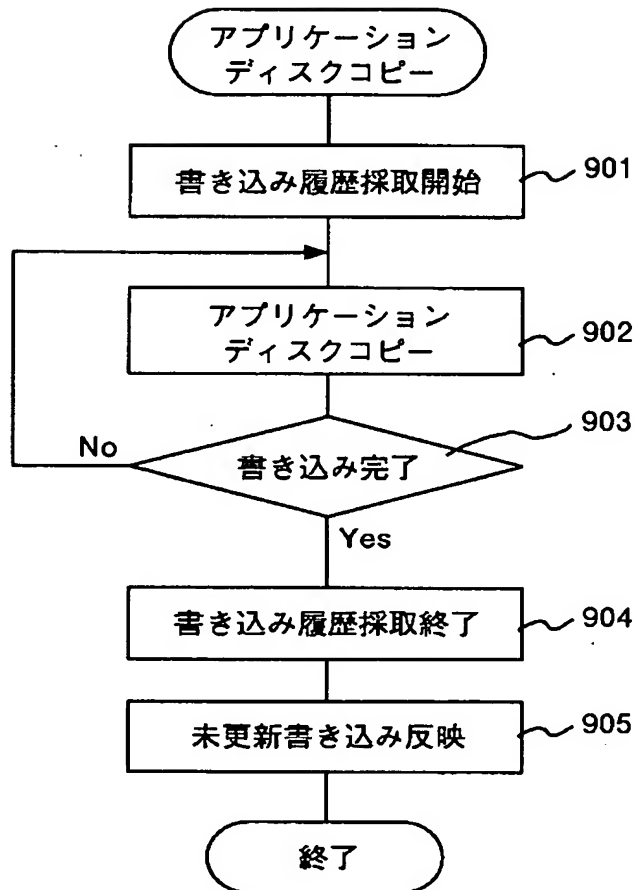
【図 8】

図 8



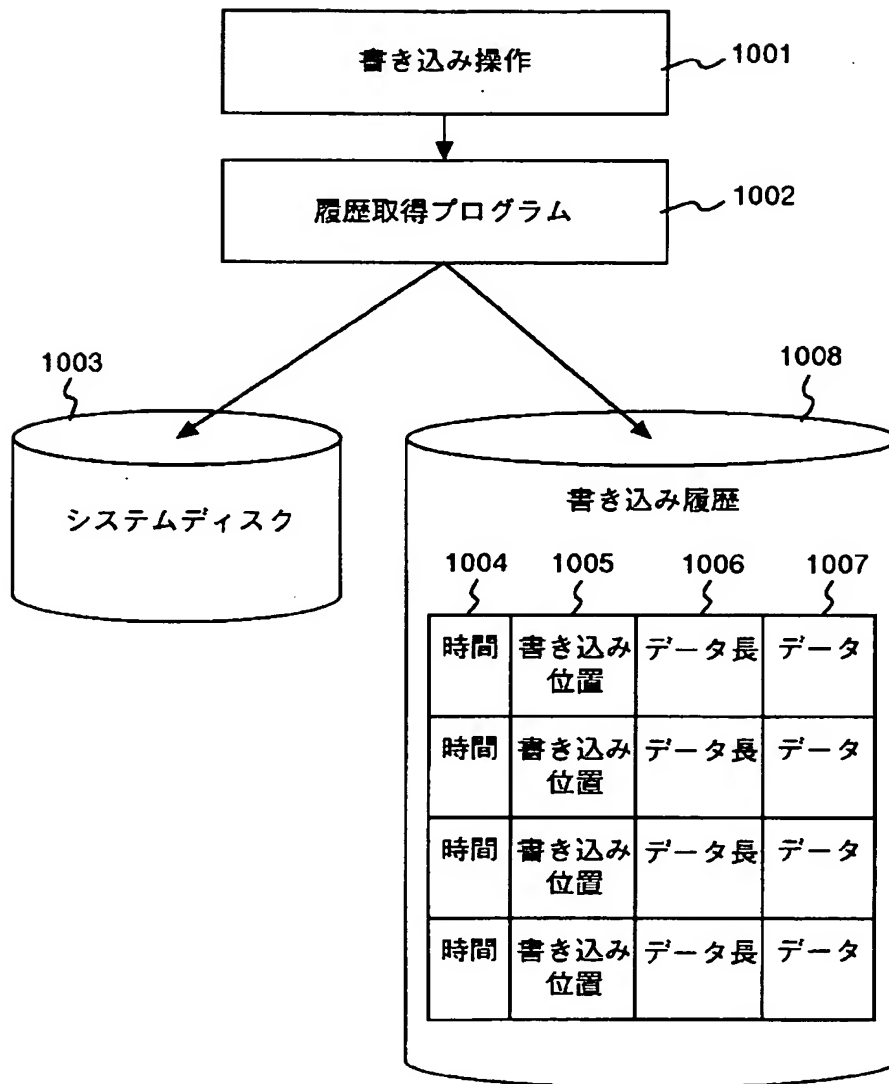
【図 9】

図 9



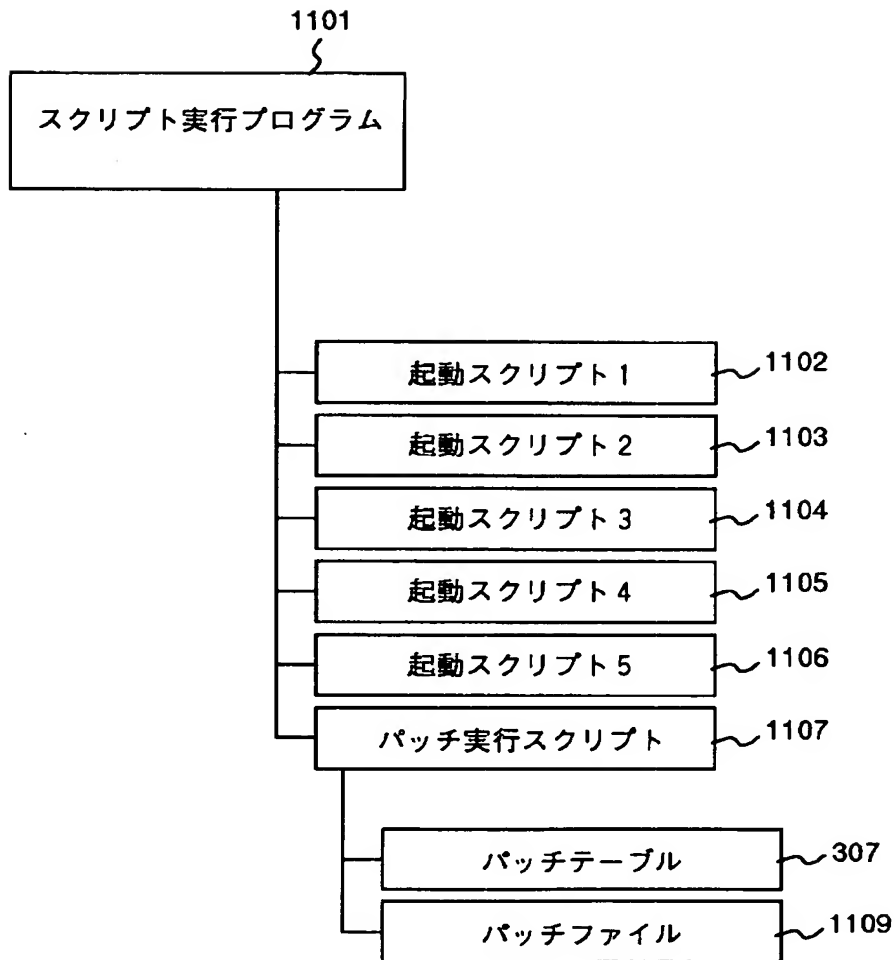
【図 10】

図 10



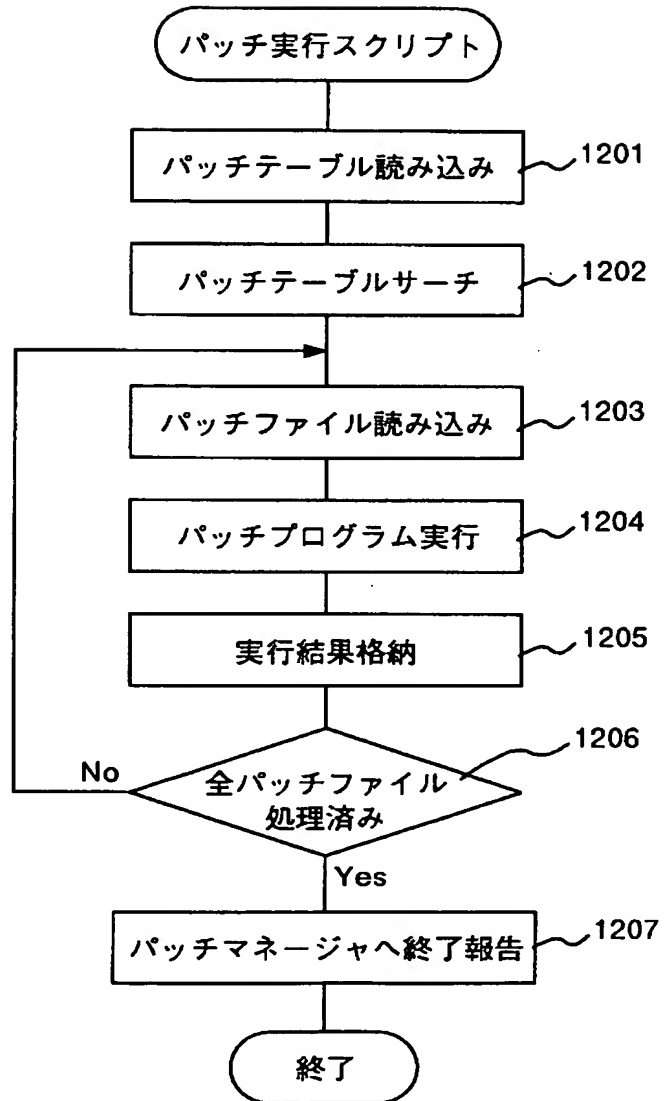
【図 11】

図 11



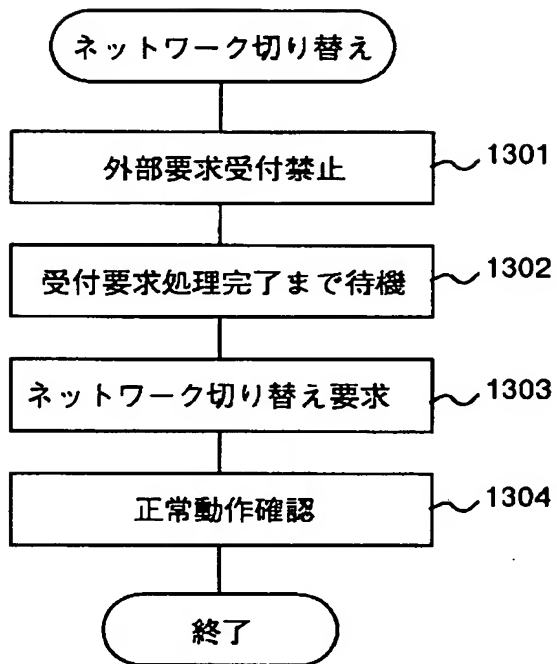
【図 1 2】

図 1 2



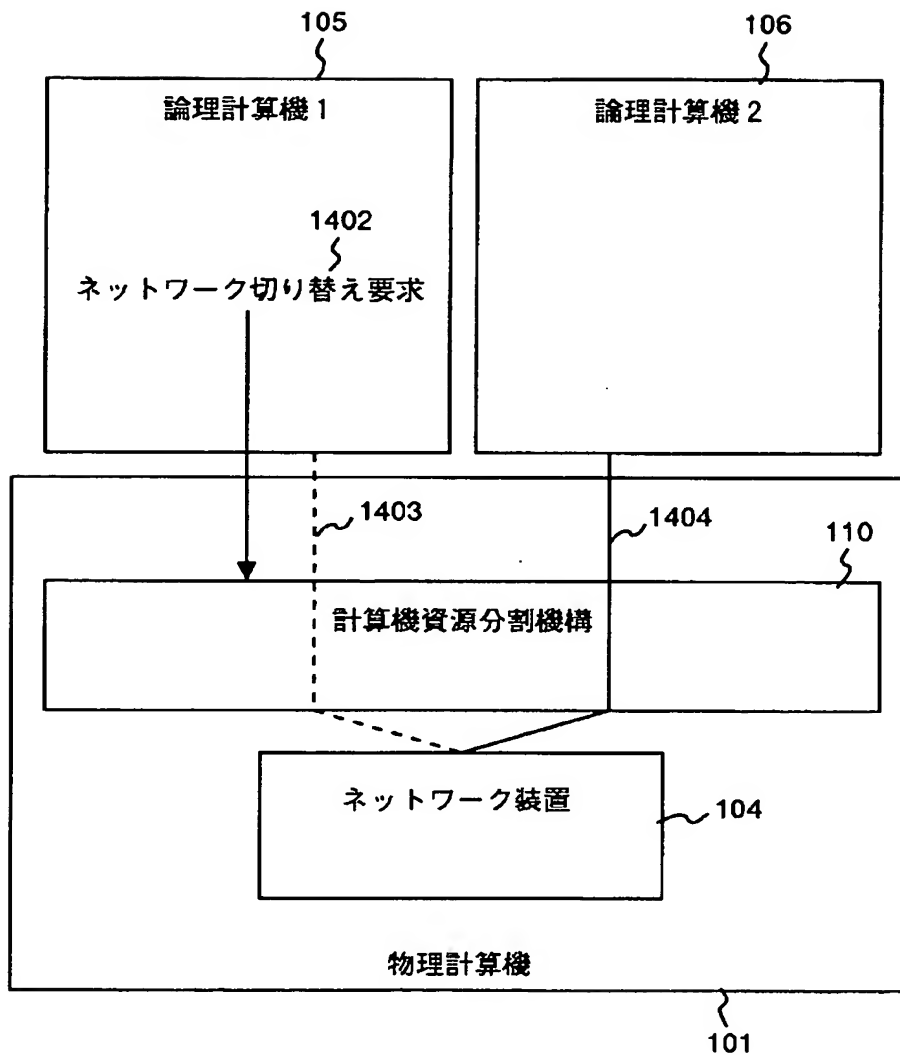
【図 1 3】

図 1 3



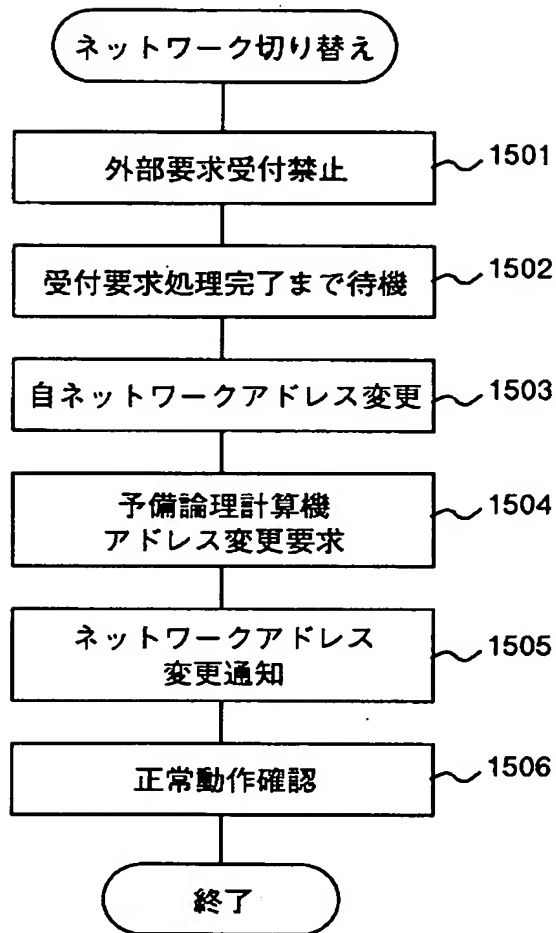
【図 1 4】

図 1 4



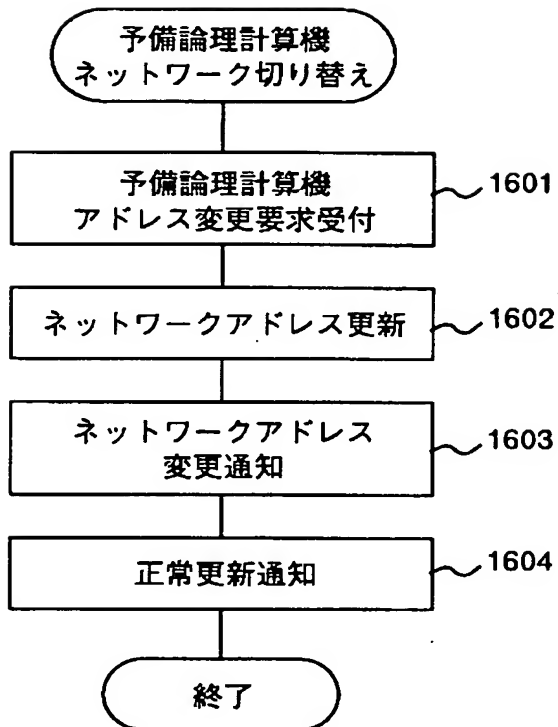
【図 1 5】

図 1 5



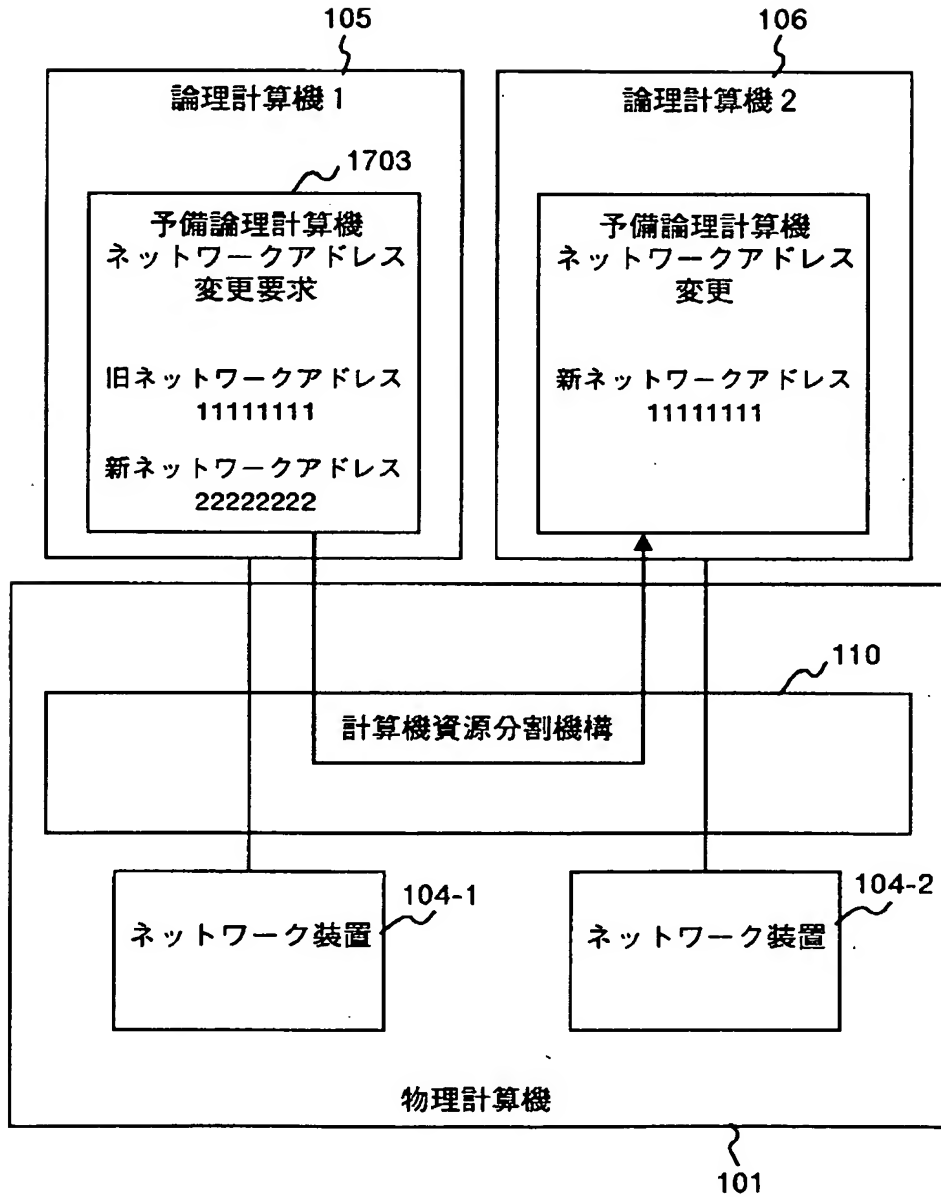
【図 1 6】

図 1 6



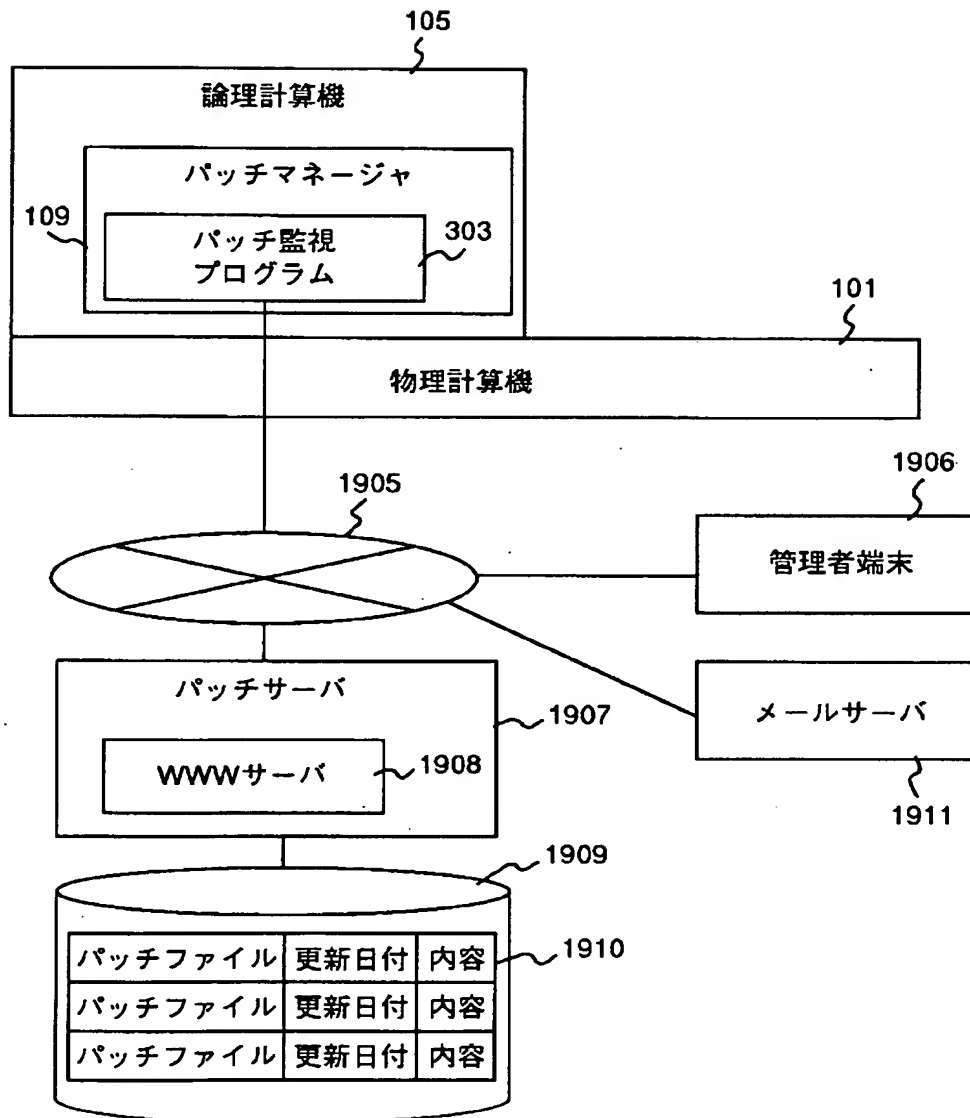
【図 17】

図 17



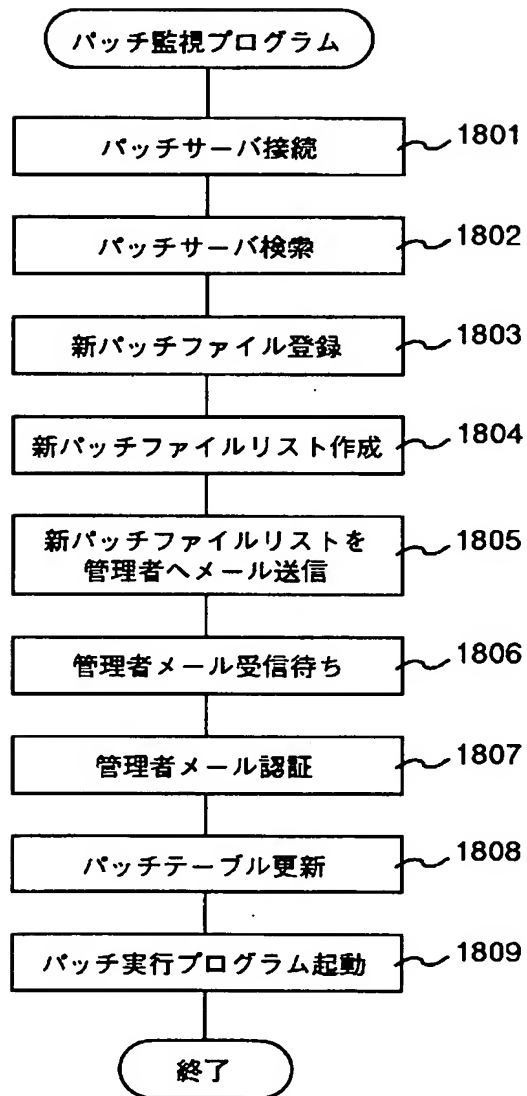
【図 18】

図 18



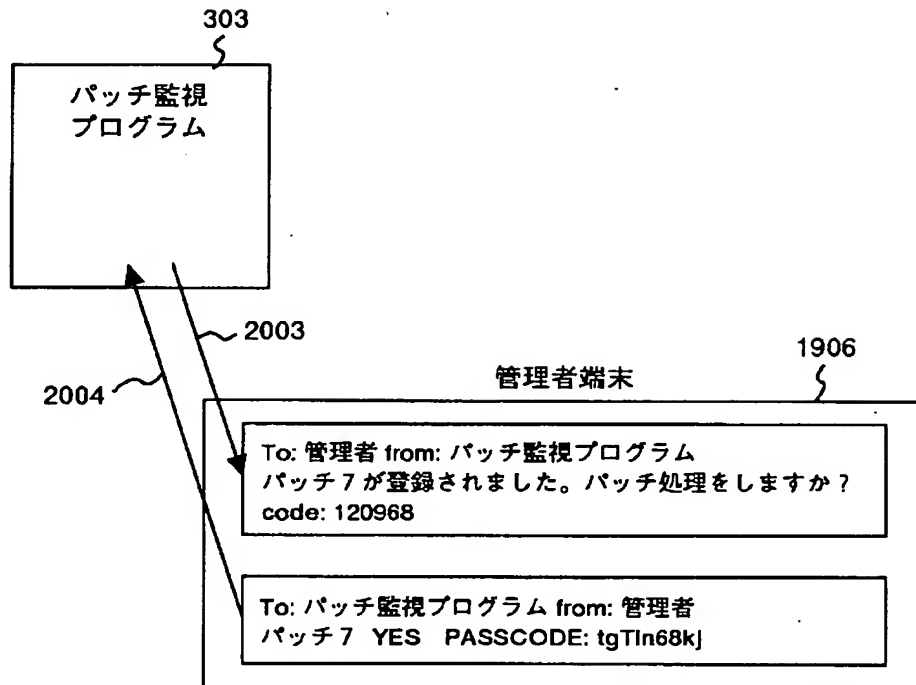
【図 1 9】

図 1 9



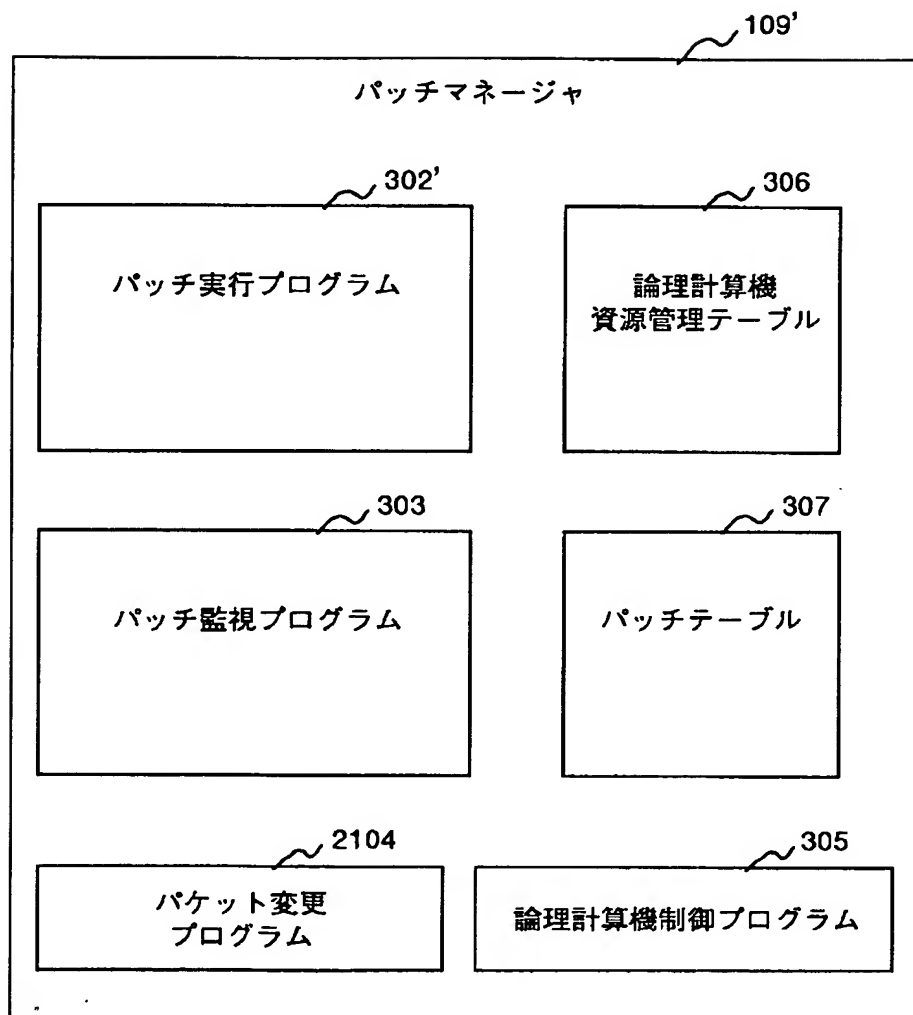
【図 2 0】

図 2 0



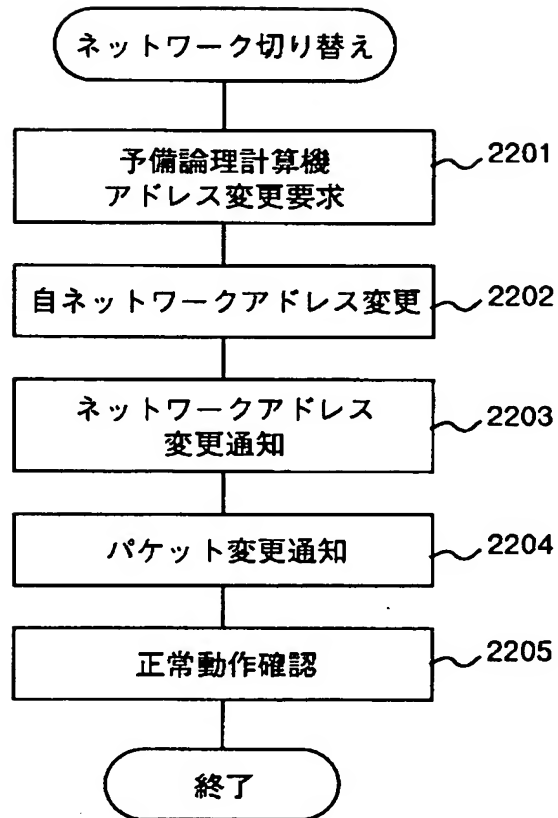
【図 2 1】

図 2 1



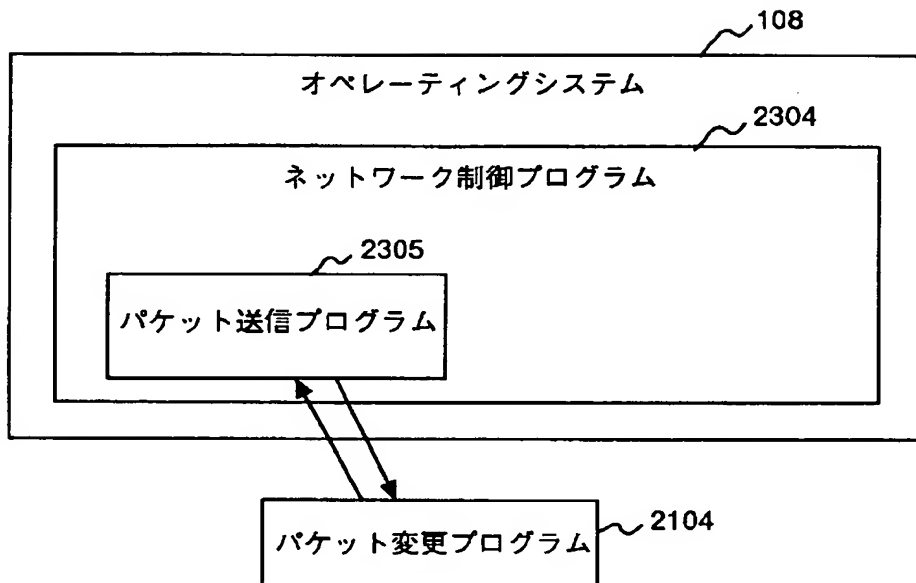
【図 2 2】

図 2 2



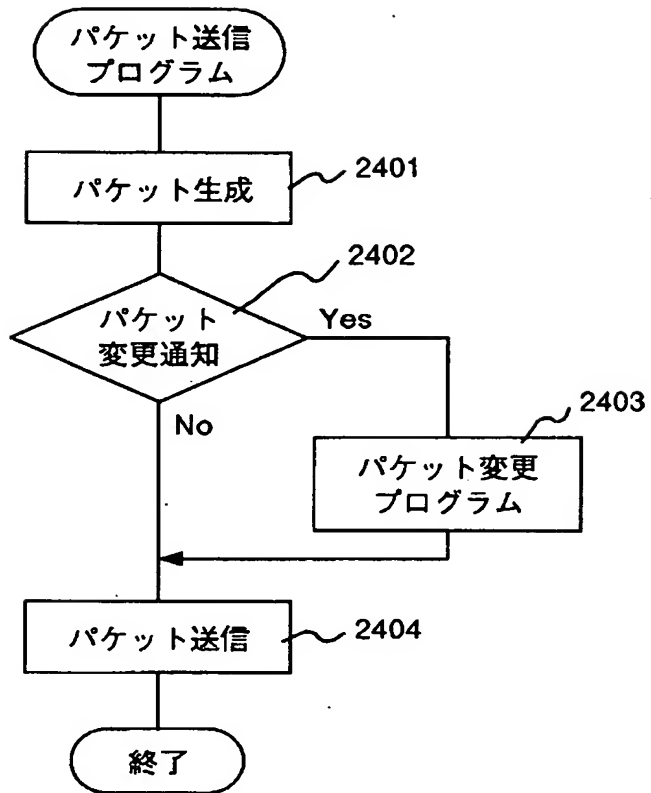
【図 2 3】

図 2 3



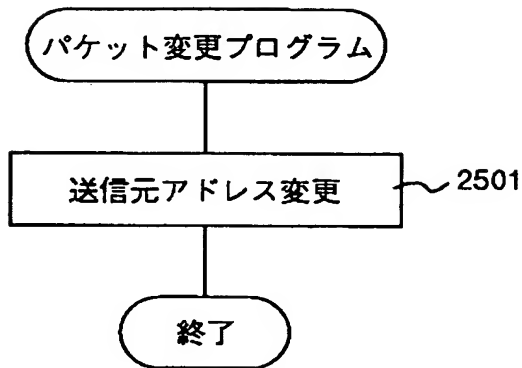
【図 2 4】

図 2 4



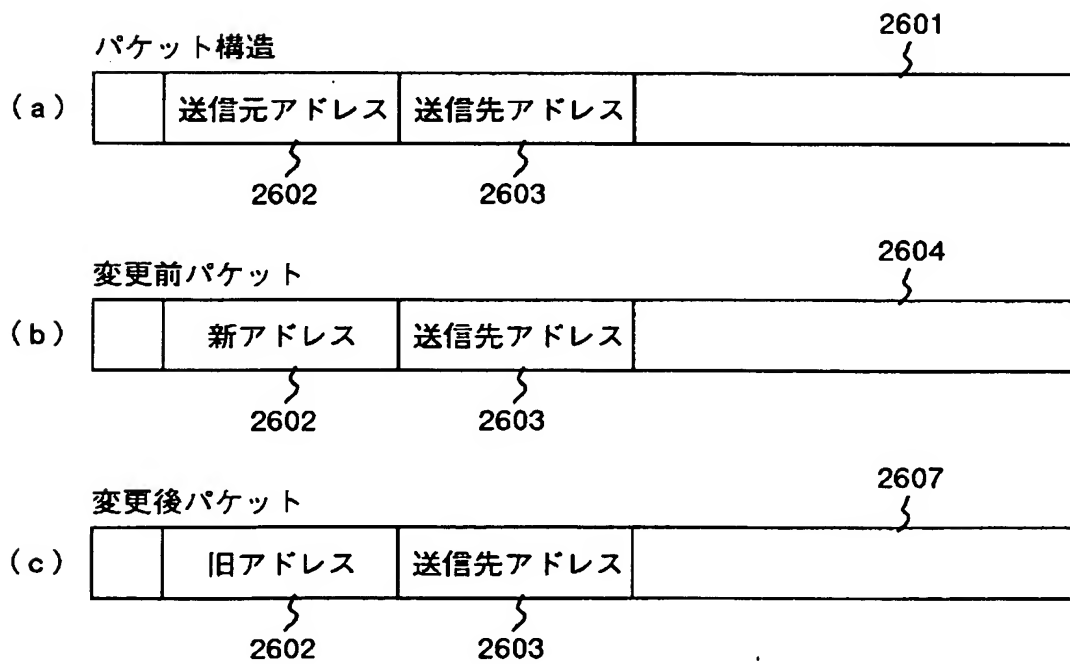
【図 2 5】

図 2 5



【図 2 6】

図 2 6



【書類名】 要約書

【要約】

【課題】 論理計算機上で稼動しているサービスを停止させることなくプログラム修正を行う。

【解決手段】 パッチマネージャ 1 0 9 は、論理計算機 1 (1 0 5) 制御下のプログラムとデータをディスクからディスクへコピーし、論理計算機 2 (1 0 6) を起動する。論理計算機 2 (1 0 6) は、コピーされたプログラム中の修正プログラムを用いてコピーされたプログラムを修正する。次にパッチマネージャ 1 0 9 は、論理計算機 1 (1 0 5) の制御下のネットワークを論理計算機 2 (1 0 6) の制御下になるように切り替えを行い、論理計算機 1 (1 0 5) の稼動から論理計算機 2 (1 0 6) の稼動に切り替える。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 1 0 8]

1. 変更年月日 1 9 9 0 年 8 月 3 1 日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台 4 丁目 6 番地

氏 名 株式会社日立製作所